대규모 데이터를 위한 *k*-최근접 이웃 학습 기반의 효율적인 협력적 여과 기법

전광성, 황규백
숭실대학교 컴퓨터학부
deltakam@naver.com, kbhwang@ssu.ac.kr

An Efficient Collaborative Filtering Method Based on *k*-Nearest Neighbor Learning for Large-Scale Data

Kwang-Sung Jun and Kyu-Baek Hwang
Division of Computing, Soongsil University

요 약

사회의 복잡화와 인터넷의 성장으로 폭발적으로 늘어나고 있는 정보들을 사용자가 모두 검토한 후 여과하기는 어려운 일이다. 이러한 문제를 보완하기 위해서 자동화된 정보 여과 기술이 사용되는데, *k*-최근접 이웃(*k*-nearest neighbor) 알고리즘은 그 구현이 간단하며 비교적 정확하여 가장 널리 쓰이고 있는 알고리즘 중 하나이다. *k* 개의 최근접 이웃들로부터 평가값을 계산하는 데 흔히 쓰이는 방법은 상관계수를 이용한 가중치에 기반하는 것이다. 본 논문에서는 이를 보완하여 대규모 데이터에 대해서도 속도는 크게 저하되지 않으며 정확도는 대폭 향상시킬 수 있는 방법을 적용하였다. 또한, 최근접 이웃을 구하는 거리함수로 다양한 방법을 시도하였다. 영화추천을 위한 실제 데이터에 대한 실험 결과, 속도의 저하는 미미하였으나 정확도에 있어서는 크게 향상된 결과를 가져올 수 있었다.

1. 서론

최근 인터넷의 발달 및 확산은 정보의 홍수라는 사회적 현상을 불러일으켰으며, 기하급수적으로 증가하는 디지털 정보를 적절히 활용할 수 있는 방안으로 협력적 여과(collaborative filtering)가 제시되어 왔다[1]. 추천시스템에서 사용되고 있는 주요한 방법인 협력적 여과는 유사한 사용자 혹은 유사한 아이템에 대하여 그 평가값들의 양상 및 성향 분석을 통해 아직 평가되지 않은 항목의 평가값을 예측한다.

협력적 여과 기법 중 *k*-최근접 이웃(*k*-nearest neighbor, *k*-NN) 학습을 이용한 방법은 보다 정교한 알고리즘들에 비해 정확도가 떨어지지만 간편한 구현과 함께 매우 빠른 학습 속도를 가지고 있기 때문에 상업적으로 널리 이용되는 기법 중 하나이다[2]. 예를 들어, 신경망(artificial neural network)이나 SVD(support vector machine)를 이용한 협력적 여과 기법들이 제안되었으나[3][4], 그 성능향상의 정도에 비해 대용량 자료에 대한 전체 수행 속도가 느리며 추천시스템이 갖추어야 할 요건 중 하나인 확장성(scalability)이 떨어지는 단점이 있다[5].

본 논문에서는 *k*-NN 학습의 장점을 유지하며 그 정확도를 향상시키는 방법을 제시한다. 보다 구체적으로, 선형회귀에 기반하여 정확도를 향상시키는 방법을 적용하였으며, 최근접 이웃 선정을 위한 거리함수로 다양한 *p*-놈(norm)을 적용하였다. 제안하는 방법은 실제 영화추천 데이터에 적용하여 그 성능을 평가하였다.

2. 선형회귀를 이용한 추천시스템

선형회귀를 이용한 *k*-NN 학습 기반의 성능 향상 기법은 [6]에서 제안되었다. 본 논문에서는 이 방법에 다양한 거리함수를 적용하여 그 성능을 향상시킨다. 우선 본 절에서는 선형회귀 기반의 추천 방법론을 기술한다.

추천을 위해 사용자 u의 아이템 i에 대한 알려지지 않은 평가점수 r_{ui}를 예측해야 한다. 최근접 이웃의 모든 접근 방법과 같이, 첫 번째 단계는 이웃의 선택이다. 본 논문에서는 아이템 기반 유사도를 사용하였다. 아이템 기반 유사도는 사용자 기반 유사도보다 높은 효율의 계산과 함께 상대적으로 높은 품질의 예측을 제공하는 것으로 알려져 있다[7]. 또한 실제로 웹사이트에 추천시스템을 적용할 때, 아이템 기반 유사도는 사용자가 학습될 필요가 없기 때문에, 새로 가입한 사용자가 몇몇 아이템만 평가하면 사용자에 대한 학습 없이 바로 추천시스템을 이용할 수 있다는 장점이 있다.

사용자 u에 의해 평가된 아이템 중에서, 아이템 i와 유사한 아이템 k개를 선정하여 이를 $N(i;u)$로 표기하는데, k 값은 보통 20-50을 사용한다[6]. 유사도 점수를 매기는 데는 상관계수를 이용하는 것이 일반적이다.

유사도의 계산은 상관계수를 이용하는 것이 일반적이지만 비슷한 성능에서 조금 더 빠른 연산을 위해 다음의 수식이 쓰일 수 있다[8]. $|U(i, j)|$는 아이템 i, j 모두를 평가한 사용자의 명수이며, α는 줄임값으로 보통 10이나 15에서 최적의 성능을 나타낸다.

$$s_{ij} = \frac{|U(i,j)|}{\sum_{u \in U(i,j)} (r_{ui} - r_{uj})^2 + \alpha} \quad (1)$$

가중치를 계산하기 위해 기존에는 상관계수를 이용한 방법이 주로 이용되었지만, 본 논문에서는 [8]에서 제안된 선형회귀 식을 푸는 최적화 알고리즘을 이용하였다. 이 기법은 다음의 식을 최소화시키는 가중치 벡터를 구한다.

$$\min_{w} \sum_{v \neq u} \left(r_{vi} - \sum_{j \in N(i;u)} w_{ij} r_{vj} \right)^2 \quad (2)$$

이 수식에서 미지수는 오직 w_{ij}들이며, 원칙적으로는 이에 대한 도함수의 선형방정식의 해를 구하여 w에 대한 최적화 문제를 풀어서 최적해를 구할 수 있다.

사용자 u를 제외한 모든 사용자 v가 모든 아이템을 평가한 이상적인 상황을 가정할 때, $A \in \mathbb{R}^{K \times K}, b \in \mathbb{R}^K$를 다음과 같이 정의할 수 있다.

$$A_{jk} = \sum_{v \neq u} r_{vj} r_{vk} \quad (3)$$

$$b_j = \sum_{v \neq u} r_{vj} r_{vi} \quad (4)$$

선형 회귀로 최적의 w_{ij}를 구하는 식은 수식 (3)과 (4)를 이용하여, 다음 방정식의 해를 구하는 것과 같다.

$$Aw = b \quad (5)$$

하지만, 실제 데이터는 희박하기 때문에, 이 식을 그대로 적용하지는 못하고, 아이템 j와 아이템 k의 지지도, 즉 두 아이템을 모두 평가한 사용자의 명수로 나눈 $\bar{A}$와 $\bar{b}$를 사용한다.

$$\bar{A}_{jk} = \frac{\sum_{v \neq u} r_{vj} r_{vk}}{|U(j,k)|} \quad (6)$$

$$\bar{b}_j = \frac{\sum_{v \neq u} r_{vj} r_{vi}}{|U(i,j)|} \quad (7)$$

수식 (6), (7)에서 지지도 값 $|U(j, k)|$가 작을 때에는 오차가 크고, 이에 따라 가중치가 지나치게 높거나 지나치게 낮게 계산되는 경우가 발생하는데, 이 문제는 가중치를 평균값 안팎으로 줄여줌으로써 완화할 수 있다. 또한 값을 줄이는 것은 선형회귀 식으로 가중치를 찾을 때, 수렴속도를 빠르게 하는 효과도 있다[8].

$$\hat{A}_{jk} = \frac{|(U(j,k)| \cdot \bar{A}_{jk} + \beta \cdot avg}{|U(j,\ k)| + \beta} \quad (8)$$

$$\hat{b}_j = \frac{|(U(i,j)| \cdot \bar{b}_j + \beta \cdot avg}{|U(i,j)| + \beta} \quad (9)$$

결국 최종적으로는 $\hat{A}_{jk}$와 $\hat{b}_j$을 이용하여 다음의 식을 푼다.

$$\hat{A}w = \hat{b} \quad (10)$$

2.1 실제 구현을 위한 전처리

본 논문에서 유사도는 앞에서 언급했던 평가 오차의 제곱의 합을 이용한 수식 (1)을 적용하였으며, 이는 예측값 계산의 효율성을 위하여 미리 계산되어 k개의 최근접 이웃을 선정하는 데 이용된다.

이와 더불어 수식 (8)의 행렬 $\hat{A}_{jk}$를 아이템의 개수 n행과 n열 배열로, 즉, 모든 아이템에 대해 생성해 놓는다. 이후 선형회귀를 이용할 때는 최근접 이웃에 대한 행만 뽑아서 $A \in \mathbb{R}^{K \times K}$와 $b \in \mathbb{R}^K$를 작성한 후 가중치 계산 알고리즘을 수행한다.

2.2 가중치 계산

$\hat{A}w = \hat{b}$ 의 선형 방정식을 풀기 위하여 *Gradient Projection* 방법을 적용하였으며[7], 그 알고리즘은 표 1과 같다. 가중치는 음수가 될 수 없다는 제약을 적용하여 음수를 허용할 때보다 더 빨리 수렴하면서도 불필요한 과대적합(overfitting)을 방지하여 정확도 및 속도 향상을 꾀하였다.

표 1. 가중치 계산 알고리즘[7]

NonNegativeQuadraticOpt($A \in \mathbb{R}^{K \times K}, b \in \mathbb{R}^K$)
% $x \geq 0$을 만족하는 $x^T Ax - 2b^T x$를 최소화한다.

do
  $r \leftarrow b - Ax$ *% residual 혹은 gradient*
% 음이 아닌 실수 제약에 걸린 active variable을 찾아 각각의 r_i를 0으로 한다.
for *i = 1, ... , k* **do**
  if $x_i = 0$ *and* $r_i < 0$ ***then***
    $r_i \leftarrow 0$
  end if
end for
  $\alpha \leftarrow \frac{r^T r}{r^T A r}$ *% 최대 스텝 크기*
% 음수값을 방지하기 위해서 스텝 크기를 조절한다.
for *i = 1, ... , k* **do**
  if $r_i < 0$ *then*
    $\alpha \leftarrow min(\alpha, -x_i / r_i)$
  end if
end for
  $x \leftarrow x + \alpha r$
while $\| r \| > \epsilon$ *% residual이 0에 가까워지면 멈춘다.*
return *x*

3. 실험 결과 및 분석

3.1 실험 환경

실험데이터는 GroupLens Research Project에서 제공한 MovieLens를 이용하였다(http://GroupLens.org). 이는 6040명의 사용자가 3883개의 영화를 평가한 약 100만개의 데이터로 구성되어 있다. 평가값은 1~5의 정수로 주어지고, 각 사용자는 최소 20개 이상의 영화를 평가하였다. 사용자와 영화에 대한 속성 정보가 주어지지만 본 논문에서는 고려되지 않았다. 정확도를 평가하기 위하여, 각 사용자의 최근 5개의 평가 데이터를 뽑아내어 학습 데

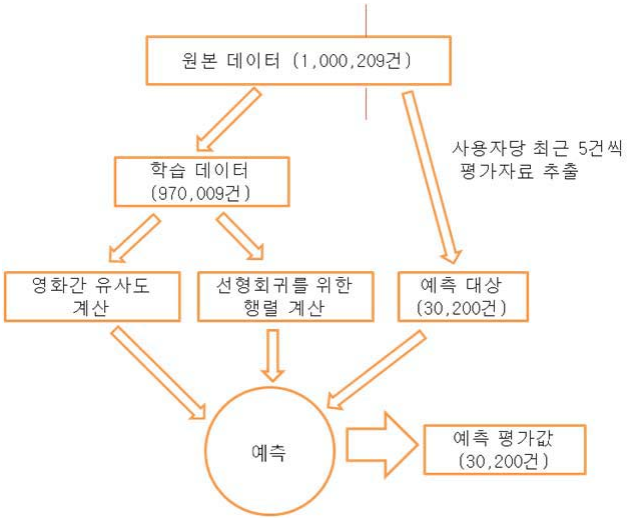

그림 1. 실험 과정 개요

이터에서 제외시켰으며, 이렇게 제외된 사용자-아이템 쌍에 대한 평가값을 예측하였다(그림 1 참조).

각 사용자에 대해 최근의 데이터를 5개 뽑아내어 시스템의 성능 평가에 활용하는 방법은 추천시스템의 취지를 잘 반영한다. 추천 시스템은 과거의 패턴에 기반하여 미래의 평가값을 예측하는 데 이용되기 때문이다.

영화에 대한 사용자의 평가값은 유사도나 선형회귀를 위한 행렬 계산을 하기 이전에 해당 영화의 평균을 뺀 후 해당 영화의 표준편차로 나누는 방법으로 정규화되었다. 이는 전체적으로 평균이 낮은 영화와 높은 영화에 대한 사람들의 선호도가 성향 파악에 공정하게 적용되도록 조절하며 유난히 사용자들이 내리는 평가의 편차가 심한 영화와 그렇지 않은 영화의 차이를 고려할 수 있도록 한다.

3.2 평가 기준

예측의 정확성을 평가하기 위하여 RMSE(root mean squared error)를 사용하였다. 이는 비슷한 측정법인 MAE(mean absolute error)보다 상대적으로 큰 오차에 민감한 특성을 갖고 있으며, 직관적으로도 사용자들은 큰 오차가 적은 빈도로 발생하기를 원하므로, 추천시스템의 성능을 평가하는데 매우 적절한 측정치이다[7]. 아래 식은 RMSE를 나타낸 것이며, 여기서 N은 총 예측 횟수, ε_i는 i번째 항목의 예측 평가값과 실제 평가값 사이의 오차이다.

$$|\bar{E}| = \sqrt{\frac{\sum_{i=1}^{N} \varepsilon_i^2}{N}} \qquad (11)$$

3.3 실험 대조군

실험의 대조군은 아이템 i와 아이템 j사이의 상관계수를 이용하여 가중치를 계산하는 방법으로 다음의 수식을 이용하였다. $\bar{r}_i$는 아이템 i에 대한 평가값들의 평균이며, $\bar{r}_j$는 아이템 j에 대한 평가값들의 평균이다.

표 2. 상관계수를 이용한 기법과 선형회귀 기반 기법의 성능 비교

이웃 개수 (k)	상관계수를 이용한 가중치 방법		선형회귀를 이용한 보간 가중치 방법	
	정확도 (RMSE)	시간 (초)	정확도 (RMSE)	시간 (초)
10	0.9186	1.8	0.9132	2.0
20	0.9144	2.2	0.9043	3.1
30	0.9155	2.6	0.9025	5.5
40	0.9175	3.1	0.9017	8.9
50	0.9176	3.4	0.9016	14.4

$$w_{ij} = \frac{\sum_v (r_{vi} - \bar{r}_i)(r_{vj} - \bar{r}_j)}{\sqrt{\sum_v (r_{vi} - \bar{r}_j)^2 \sum_v (r_{vj} - \bar{r}_j)^2}} \qquad (12)$$

이렇게 구한 w_{ij}는 사용자 u가 아이템 i에 대해 매긴 평가값의 예측치 r_{ui}를 구하는 데 이용되며, 다음의 식을 통해 계산된다[9].

$$r_{ui} = \bar{r}_i + \frac{\sum_j w_{ij} (r_{uj} - \bar{r}_j)}{\sum_j w_{ij}} \qquad (13)$$

3.4 실험 결과

실험 결과는 표 2 및 그림 2에 정리되어 있다. 최근접 이웃의 개수를 10, 20, 30, 40, 50으로 증가시키며 알고리즘을 적용했고, 시간은 같은 실험을 10회 반복한 후 평균을 내어 계산하였다.

실험에 사용된 파라미터는, 유사도 수식 (1)의 α값은 15, 선형회귀에 이용되는 수식 (8)의 행렬 $\hat{A}$의 β값은 500으로 설정하였다. 파라미터의 값을 정할 때는 α와 β가 서로 독립이라는 가정 하에 실험을 진행하였다. 먼저 α와 β를 모두 1로 정하고 나서, β값을 5, 10, 25까지 증가시키고 그 이후로는 두 배씩 증가시켜 실험을 진행했으며, β값에 따른 정확도가 상승하다가 감소하는 시점을 발견하면, 그 정점을 최적의 파라미터값으로 정의했다. β 값을 정한 후 α값에 대해서는 1, 3, 5, 10, 15, 20, 25를 차례로 대입하여 그 최적의 값을 검색하였다.

파라미터 α는 그 값의 차이에 따른 민감도가 크지 않았던 반면, 파라미터 β는 그 값의 차이에 따른 정확도의 차이가 매우 심하였다. 예를 들어, β값을 1로 두고 k를 20으로 하여 실험을 진행하였을 경우 RMSE는 0.9890으로 다른 파라미터 값들의 경우와는 크나큰 차이를 보였다.

표 2를 살펴보면 선형회귀를 이용한 보간 가중치를 이용했을 경우에 상관계수를 이용하는 경우 보다 더욱 높은 정확도를 보인다는 사실을 알 수 있다.

또 한가지 주목할 점은 이웃 k의 개수가 증가할 때, 상관계수를 이용한 가중치를 이용하면 정확도가 오히려 떨어지는 현상이다. 따라서 상관계수의 경우 최적의 k값

을 정하는 수고가 필요하다. 반면, 선형회귀를 이용한 방법은 이웃의 개수가 늘어날수록 정확도가 향상되는 것을 볼 수 있다(그림 2 참조). 따라서 k값에 대한 고민을 하지 않아도 되며, 추천시스템을 구현할 경우 계산자원에 여유가 있다면, 필요한 만큼 이웃의 개수를 늘려 성능향상을 도모할 수 있다고 여겨진다.

반면 시간은 100만 건 기준, 이웃 개수 $k = 20$을 기준으로 할 때, 상관계수를 이용한 방법은 2.2초 소모되었으며, 선형회귀를 이용한 보간 가중치 방법은 3.1초 소모되었다. 한 건당 시간으로 환산하면, 상관계수를 이용한 방법이 0.0000022초, 0.0000031초로 수행속도는 조금 저하되는 것을 확인할 수 있다.

선형회귀를 이용한 방법이 정확도가 향상된 이유는 다음과 같이 해석될 수 있다. 먼저, 상관계수를 이용한 방법은 단 두 가지 아이템만 놓고 값들을 비교한 수치의 비율을 직접 가중치에 적용시켰기 때문에 실제 수십 개의 이웃 아이템들이 예측값에 함께 어떤 영향을 미칠 수 있는지에 대해서는 고려하지 않게 된다.

반면, 선형회귀를 이용한 방법은 예측하고자 하는 아이템에 대해 그것과 유사한 이웃들을 동시에 고려하여 최적의 가중치를 찾아내기 때문에, 높은 정확도를 보이는 것으로 볼 수 있다. 또한, 가중치의 총 합이 1로 제약되지 않는 것도 이웃의 평가값들과의 상호 작용에 유연성을 더하게 된다고 생각된다.

상관계수를 이용한 방법에서 이웃의 개수가 증가함에 따라 정확도가 떨어진 이유는 실제로 상관관계가 떨어지는 평가값들도 비율상 일정부분 반영이 되기 때문이다. 반면, 선형회귀를 이용한 방법은 경험적으로 상관관계가 없다고 판단되는 평가값에는 0의 가중치가 부여될 수 있으므로, 이웃이 많을수록 잠재적으로 영향이 있는 아이템을 예측에 반영하면서 정확도의 증가를 가져온다.

3.5 새로운 유사도 함수에 대한 실험

사용자의 성향을 잘 반영하는 유사도 공식을 정의하는 것은 추천시스템에 있어 매우 중요한 역할을 하게 된다. 기존에 이용되던 수식 (1)과 같은 유사도는 대부분 값이 1 이하로 작아서 아이템간의 유사도를 잘 표현하지 못하고 있다는 판단 아래, 새로운 유사도 식에 대한 실험을 진행하였다. 처음에는 다음과 같이 분모에 p-놈(norm)을 적용하고자 시도하였으나, 분모가 지나치게 작아지면서 역으로 유사도 값이 너무 커져서, 아이템간의 관계를 잘 표현하지 못하게 되었다.

$$s_{ij} = \frac{|U(i,j)|}{\left\{\sum_{u \in U(i,j)} |r_{ui} - r_{uj}|^p\right\}^{\frac{1}{p}} + \alpha} \qquad (14)$$

따라서 분모의 값이 지나치게 커지지 않도록 p-놈에서 $1/p$승을 계산하는 부분을 제외시킨 다음의 식에 대해 실험을 다시 진행하였다.

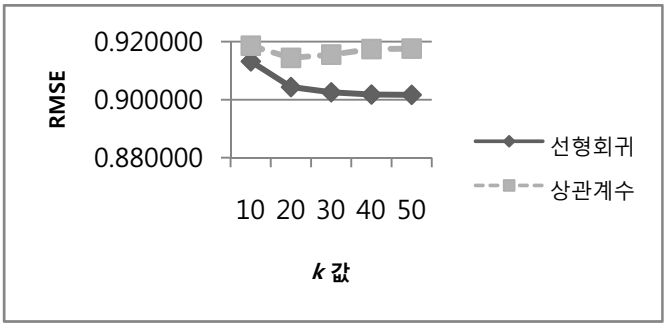

그림 2. k 값의 차이에 의한 RMSE 변화 추이

$$s_{ij} = \frac{|U(i,j)|}{\sum_{u \in U(i,j)} |r_{ui} - r_{uj}|^p + \alpha} \qquad (15)$$

앞에서 보여주었던 유사도 식과 유사하지만, 사용자 u의 아이템 i에 대한 평가값과 사용자 u의 j에 대한 평가값의 오차에 제곱을 하는 것이 아니라 p승을 하였다.

실험은 이웃의 개수 k를 20으로 고정시키고, p의 값은 0.2부터 3.0까지 0.2단위로 증가시키되 2.0을 제외하고 시도하였으며, 줄임값 α는 5, 10, 15를 기본으로 실험하고, 이 중 정확도의 최소값이 5일 경우 3과 1, 0.5를 검사하여 최소값을 찾았고, 정확도의 최소값이 15일 경우 20과 25를 검사하여 최소값을 찾는 방법을 이용하였다. p 값이 작은 경우 데이터의 잡음의 영향을 감쇄할 수 있는 효과가 있다.

실험결과는 표 3과 같았으며, p값이 1.4, α값이 5 일 때 RMSE가 0.9039로 표 2에서 관찰된 기존의 유사도 방법을 이용한 경우의 RMSE 0.9043보다 약간의 정확도 향상이 있었다. 그러나, 향상 정도가 그리 크지 않은데다, 유사도의 파라미터 α값에 대한 실험도 5와 10사이의 값에 대한 실험치는 현재는 없는 관계로 향후 조금 더 정밀한 실험이 필요할 것으로 생각된다.

표 3. 다양한 유사도 실험 결과 ($k = 20$)

p 값	RMSE	최적의 α
0.2	0.9180	1
0.4	0.9136	5
0.6	0.9099	1
0.8	0.9072	10
1.0	0.9049	5
1.2	0.9044	5
1.4	**0.9039**	**5**
1.6	0.9042	10
1.8	0.9044	10
2.2	0.9046	15
2.4	0.9044	20
2.6	0.9050	20
2.8	0.9064	15
3.0	0.9068	20

4. 결론

k-최근접 이웃 알고리즘은 높은 정확도의 신경망이나 SVD 알고리즘보다 간단 명료하고 빠른 학습 속도를 갖추고 있어 추천시스템 구현에 널리 적용되고 있는 알고리즘 중 하나이다. 그러나, *k*개의 최근접 이웃들로부터 평가값을 계산하는 데 흔히 쓰이는 상관계수를 이용한 방법은 정확도가 지나치게 떨어지는 단점이 존재한다. 본 논문에서는 선형회귀를 이용하여 가중치를 계산하는 방법이 기존의 상관계수 방법보다 향상된 정확도를 보이면서도 수행 속도는 크게 저하되지 않는다는 사실을 실제 대규모 영화추천 데이터에 대해 보였다. 또한, 더 나은 정확도를 제공할 수 있는 아이템 기반 유사도를 적용하여, 정확도를 더욱 향상시킬 수 있었다.

참고문헌

[1] J. Schafer, Recommender systems in e-commerce, *Proceedings of the First ACM Conference on Electronic Commerce*, pp. 158-166, 1999.

[2] 박지선, 김택헌, 류영석, 양성봉, 추천 시스템을 위한 2-way 협동적 필터링 방법을 이용한 예측 알고리즘, *정보과학회논문지 : 소프트웨어 및 응용*, 제29권, 제9호, pp. 669-675, 2002.

[3] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, Incremental SVD-based algorithms for highly scalable recommender systems, *Proceedings of the Fifth International Conference on Computer and Information Technology*, 2002

[4] C. Christakou and A. Stafylopatis, A hybrid movie recommender system based on neural networks, *Proceedings of the Fifth International Conference on Intelligent Systems Design and Applications*, pp. 500-505, 2005

[5] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, GroupLens: an open architecture for collaborative filtering of netnews, *Proceedings of the ACM 1994 Conference on Computer Supported Cooperative Work*, pp. 175-186, 1994.

[6] R. Bell and Y. Koren, Improved neighborhood-based collaborative filtering, *Proceedings of the 2007 KDD Cup and Workshop*, 2007.

[7] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, Itembased collaborative filtering recommendation algorithms, *Proceedings of the 10th International Conference on the World Wide Web*, pp. 285-295, 2001.

[8] R. Bell, Y. Koren, and C. Volinsky, Modeling relationships at multiple scales to improve accuracy of large recommender systems, *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 95-104, 2007.

[9] J. Breese, D. Heckerman, and C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pp. 43-52, 1998.